

AD-A057 872

SYRACUSE UNIV N Y

F/G 9/2

BAYESIAN SOFTWARE PREDICTION MODELS. VOLUME III. AVAILABILITY A--ETC(U)

JUL 78 K OKUMOTO, A L GOEL

F30602-76-C-0097

UNCLASSIFIED

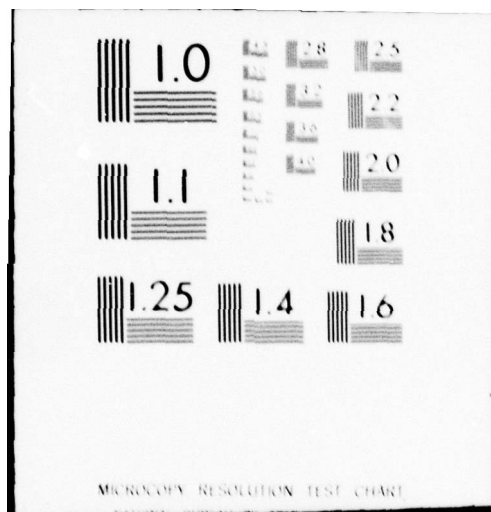
TR-78-3

RADC-TR-78-155-VOL-3

NL

1 OF 1
AD
A057 872





ADA057872

AD No. _____
DDC FILE COPY

② LEVEL III

RADC-TR-78-155, Volume III (of five)
Final Technical Report
July 1978

A057870
A057873

A057871



BAYESIAN SOFTWARE PREDICTION MODELS
Availability Analysis of Software Systems
Under Imperfect Maintenance

K. Okumoto
Amrit L. Goel

Syracuse University

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

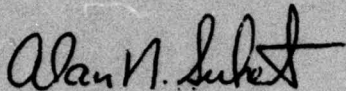
DDC
RECEIVED
AUG 23 1978
B

78 08 22 062

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

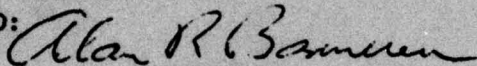
RADC-TR-78-155, Volume III (of five) has been reviewed and is approved for publication.

APPROVED:



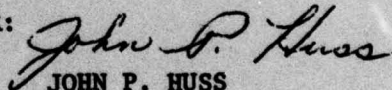
ALAN N. SUKERT
Project Engineer

APPROVED:



ALAN R. BARNUM
Assistant Chief
Information Sciences Division

FOR THE COMMANDER:



JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

(19) TR-78-155-VOL-3

(18) RADC

UNCLASSIFIED

(14) TR-78-3

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
RADC-TR-78-155, Vol III (of five)		
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
BAYESIAN SOFTWARE PREDICTION MODELS. Volume III. Availability Analysis of Software Systems Under Imperfect Maintenance.	Final Technical Report, Dec 75 - Mar 78	
6. AUTHOR(s)	7. PERFORMING ORG. REPORT NUMBER	8. CONTRACT OR GRANT NUMBER(s)
K. Okumoto Amrit L. Goel	Technical Report No. 78-3	F30602-76-C-0097
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Syracuse University Syracuse NY 13210		62702F 55811403
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Rome Air Development Center (ISIS) Griffiss AFB NY 13441		July 1978
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES
Same		41
15. SECURITY CLASS. (of this report)		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
UNCLASSIFIED		N/A
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
Same		
18. SUPPLEMENTARY NOTES		
RADC Project Engineer: Alan N. Sukert (ISIS)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Imperfect Debugging Software Maintenance Software Error Prediction Software Error Modeling Software Availability		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>In this report we developed a model for the operational phase of a software system, which incorporates the uncertainty of error removal and the time spent in correcting errors. Expressions for various measures of software system performance, e.g. distribution of time to a specified number of remaining errors, the expected number of errors detected and corrected by time t, and software system availability are derived. Numerical examples are used to illustrate these results.</p> <p>Volume V will be published at a later date.</p>		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

339 600 78 08 22 062

Full

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block 7.

¹Research Assistant, Department of Industrial Engineering
and Operations Research, Syracuse University.

²Professor, Department of Industrial Engineering and Operations
Research, and School of Computer and Information Science, Syracuse
University.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

1.	INTRODUCTION	1
2.	MODEL DEVELOPMENT.	3
3.	DERIVATION OF VARIOUS QUANTITIES OF INTEREST	11
3.1	Distribution of Time to a Specified Number of Remaining Errors	11
3.1.1	Mean and variance of the first passage time	14
3.1.2	Numerical example.	14
3.2	State Occupancy Probabilities and Software System Availability	18
3.3	Number of Software Errors Detected by Time t . . .	22
3.4	Number of Software Errors Corrected by Time t . .	23
4.	GAMMA APPROXIMATION FOR A LARGE-SCALE SOFTWARE SYSTEM. .	26
5.	CONCLUDING REMARKS	33
6.	SELECTED REFERENCES.	34
	APPENDIX A	37

REVISION No.	
NYO	NYO Section <input checked="" type="checkbox"/>
SEC	SEC Section <input type="checkbox"/>
UNCLASSIFIED	<input type="checkbox"/>
IDENTIFICATION	
DISTRIBUTION/AVAILABILITY CODES	
AVAIL. and/or SPECIAL	
A	

LIST OF FIGURES

Figure	Page
2.1 A Diagrammatic Representation of Transitions Between States of $X(t)$	5
2.2 A Typical Realization of the $X(t)$ Process.	8
3.1 PDF of First Passage Time From N to n_0	15
3.2 CDF of First Passage Time From N to n_0	16
3.3 Software System Availability and State Occupancy Probabilities at Time t	21
3.4 Expected Number of Software Errors Detected and Corrected by Time t	25
4.1 Relative Losses for the Third and Fourth Moments . .	30
4.2 First Passage Times Based on Gamma Approximation . .	31
4.3 Plots of State Occupancy Probabilities and Software System Availability Based on Gamma Approximation . .	32

LIST OF TABLES

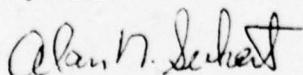
Table		Page
3.1	Mean and Variance of First Passage Time for Various n_0	17
4.1	Gamma Approximations for First Passage Time Distributions.	29

EVALUATION

The necessity for more complex software systems in such areas as command and control and avionics has led to the desire for better methods for predicting software errors to insure that software produced is of higher quality and of lower cost. This desire has been expressed in numerous industry and Government sponsored conferences, as well as in documents such as the Joint Commanders' Software Reliability Working Group Report (Nov 1975). As a result, numerous efforts have been initiated to develop and validate mathematical models for predicting such quantities as the number of remaining errors in a software package, the time to achieve a desired availability level, and a measure of the software availability. However, early efforts have not produced models with the desired accuracy of prediction and with the necessary confidence limits for general model usage.

This effort was initiated in response to this need for developing better and more accurate software error prediction models and fits into the goals of RADC TPO No. 5, Software Cost Reduction (formerly RADC TPO No. 11, Software Sciences Technology), in the subthrust of Software Quality (Software Modeling). This report summarizes the development of a mathematical model for predicting quantities, such as the expected number of remaining errors, achieved availability, and time to detect and correct a specified number of errors, for operational software systems that assumes a software error is not corrected at a given time with probability 1 (i.e. imperfect debugging). The importance of this development is that it represents the first attempt to develop software error prediction models that incorporate imperfect debugging, and thus more closely reflect the actual software error detection and correction process.

The theory and equations developed under this effort will lead to much needed predictive measures for use by software maintenance personnel in providing better and more efficient maintenance of operational software. In addition, the associated confidence limits and other related statistical quantities developed under this effort will insure more widespread use of these modeling techniques. Finally, the predictive measures and equations developed under this effort will be applicable to current Air Force software development projects and thus help to produce the high quality, low cost software needed for today's systems.



ALAN N. SUKERT
Project Engineer

1. INTRODUCTION

Considerable emphasis has been placed in recent years on empirical studies of software error phenomena with the objective of improving software performance. Such studies can be classified into one (or both) of two categories. In the first category the emphasis is on the analysis of software error data collected from small or large projects, during development and/or operational phases. Studies in the second category are primarily aimed at the development of analytical models which are then used to obtain the reliability and other quantitative measures of software performance.

Typical of the first category are the studies by Akiyama [1], Belady and Lehman [3], Fries [6], Endres [5], Baker [2], Motley et al [16], Miyamoto [14], Willman et al [31], Schneidewind [22], Shooman et al [25], Sukert [26,27], Rye et al [20], Thayer et al [28], and Wagoner [30]. These studies range in size from an analysis of small data sets (108 errors), e.g. Wagoner [30], to analysis of large sets (3500 errors), e.g. Thayer et al [28] and encompass data from an on-line system [14], an operating system [3], to that from the Apollo project [20].

In the second category of papers, several models have been proposed and studied during the last six years. These include 'exponential type' models of Shooman [24], Jelinski and Moranda [10,11], and Schick and Wolverton [21]; models based on the non-homogeneous Poisson process proposed by Goel and Okumoto [8] and

Schneidewind [23], and a Bayesian model by Littlewood and Verrall [13]. Halstead [9] has developed a theory based on 'software physics' for various measures of the performance of a software system. Musa [17] has introduced a model which is based on a large number of parameters derived from the software system being modelled. Trivedi and Shooman [29] consider a Markov model in which they incorporate the time spent for removal of errors.

Most of the above studies assume that an error is removed with certainty when detected. Goel and Okumoto [7] have developed a model for the debugging phase which takes into consideration the uncertainty of error removal. Using this model they have derived expressions for various quantities of interest. However, they assume that the time for error removal is negligible.

In this report we present a model for the operational phase of the system for the case when errors are not removed with certainty and also take into consideration the time spent for error removal. Expressions for various quantities of interest, e.g. distribution of time to a specified number of remaining errors, expected number of errors detected and corrected by time t , and software system availability, are derived from this model. The basic model is developed in Section 2, and the quantities of interest are derived in Section 3. Approximations for large-scale software systems using a Gamma distribution are discussed in Section 4.

2. MODEL DEVELOPMENT

The process to be modelled consists of a sequence of operational and maintenance (up and down) states of the software system. We make the following assumptions about the process and the software system.

- (i) The errors in the software system are independent of each other and have a constant occurrence rate λ .
- (ii) The probability of two or more errors occurring simultaneously is negligible.
- (iii) When the system is inoperative due to the occurrence of an error, the error causing the failure, when detected, is corrected with probability p ($0 \leq p \leq 1$) while with probability q ($p+q=1$) the error is not removed. Thus q is the probability of imperfect maintenance.
- (iv) The time to remove an error when there are i remaining errors in the system, Y_i , follows an exponential distribution with parameter μ_i .
- (v) No new errors are introduced during the error removal (correction) phase.
- (vi) At most one error is removed at correction time.

Let $X(t)$ denote the state of the system at time t . Define

$$X(t) = \begin{cases} i; \text{ The software system is operational while} \\ \quad \text{there are } i \text{ errors remaining in the} \\ \quad \text{software system, } i=0,1,2,\dots,N. \\ D; \text{ The software system is down for error} \\ \quad \text{removal (maintenance).} \end{cases} \quad (2.1)$$

We will use this random variable to describe the state of the system at time t . Further, let N be the number of errors at the beginning of the operational phase, i.e., $X(0) = N$.

Suppose the system is operative with i remaining errors when a failure occurs. Further, suppose that the error removal activity (maintenance) is carried on up to time t . Then, from assumption (iii) we have

$$X(t) = \begin{cases} i-1 & \text{with probability } p \\ i & \text{with probability } q. \end{cases} \quad (2.2)$$

In other words, if we were to observe the $X(t)$ process at the end of each maintenance phase, then its behavior is governed by (2.2). It should be noted that in making these transitions $X(t)$ always goes through the D state as defined in (2.1). A diagrammatic representation of transitions between states $N, N-1, \dots, 1, 0$ and D is given in Figure 2.1. In general, the transition probabilities $P_{ij}^{(D)}$ from state i to state j via D , $i, j = 0, 1, 2, \dots, N$ are given by

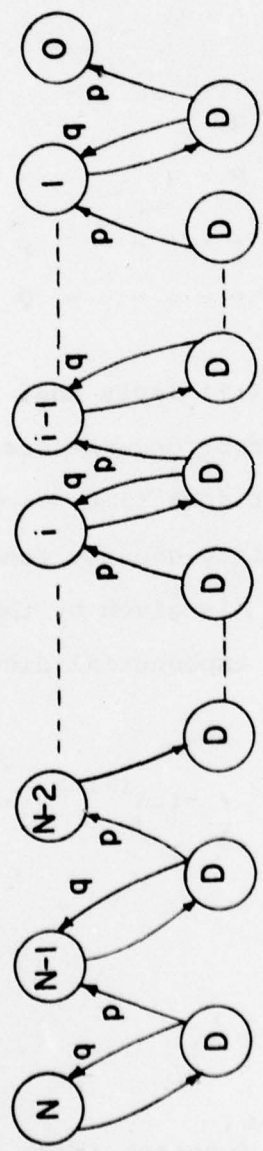


Figure 2.1 A Diagrammatic Representation of Transitions Between States of $X(t)$

$$[P_{ij}^{(D)}] = \begin{matrix} & \begin{matrix} 0 & 1 & 2 \dots i-2 & i-1 & i & \dots & N-2 & N-1 & N \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ i-1 \\ i \\ \vdots \\ N-1 \\ N \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & - & - & - & - & - & 0 & 0 \\ p & q & 0 & - & - & - & - & - & 0 & 0 \\ 0 & p & q & & & & & & | & | \\ \vdots & \vdots & \vdots & \diagdown & \diagdown & \diagdown & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & p & q & & & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & p & q & & \vdots & \vdots \\ 0 & 0 & - & - & - & - & - & p & q & 0 \\ 0 & 0 & - & - & - & - & - & 0 & p & q \end{bmatrix} \end{matrix} \quad (2.3)$$

Now, assumptions (i) and (ii) imply that the times between successive software failures (error occurrences) follow an exponential distribution. Suppose at some time $t = \tau$, $x(\tau) = i$, $i = 0, 1, \dots, N$. Then the probability density function (pdf) $f_i(t)$ of the time to next failure, T_i , is given by the distribution of the first order statistic of i exponential distributions each with parameter λ , i.e.,

$$f_i(t) = \binom{i}{1} \lambda e^{-\lambda t} \cdot (e^{-\lambda t})^{i-1}$$

or

$$f_i(t) = i\lambda \cdot e^{-i\lambda t}$$

or

$$f_i(t) = \lambda_i e^{-\lambda_i t} \quad (2.4)$$

and the cumulative distribution function (cdf) is given by

$$F_i(t) = 1 - e^{-\lambda_i t} \quad (2.5)$$

where $\lambda_i = i\lambda$.

It should be pointed out that under assumptions (i) and (ii), we take the parameter λ_i to be equal to $i\lambda$. However, in general under different assumptions λ_i could be some other function of i and λ .

From assumption (iv), the cdf of maintenance times is obtained as

$$P(Y_i \leq t) = 1 - e^{-\mu_i t}. \quad (2.6)$$

Let Z_i denote the time for one up-down cycle when the number of remaining errors is equal to i , i.e.,

$$Z_i = T_i + Y_i. \quad (2.7)$$

Then

$$f_i(t) = P(Z_i \leq t) = \left(1 - e^{-\lambda_i t}\right) * \left(1 - e^{-\mu_i t}\right) \quad (2.8)$$

where $*$ denotes convolution.

Now we note that even though the stochastic process $X(t)$ makes transitions from state to state in accordance with equation (2.3), the times spent in various states are random and are given by equation (2.8). Hence $\{X(t), t \geq 0\}$ forms a semi-Markov process. A typical realization of this process is shown in Figure 2.2. It should be pointed out that in our formulation the process $X(t)$ undergoes both real and virtual transitions. This means that after an attempt to remove an error the state of $X(t)$ may change or may remain unchanged. In Figure 2.2 real transitions occur at states $N, N-2$ and i while a virtual transition occurs at state $N-1$.

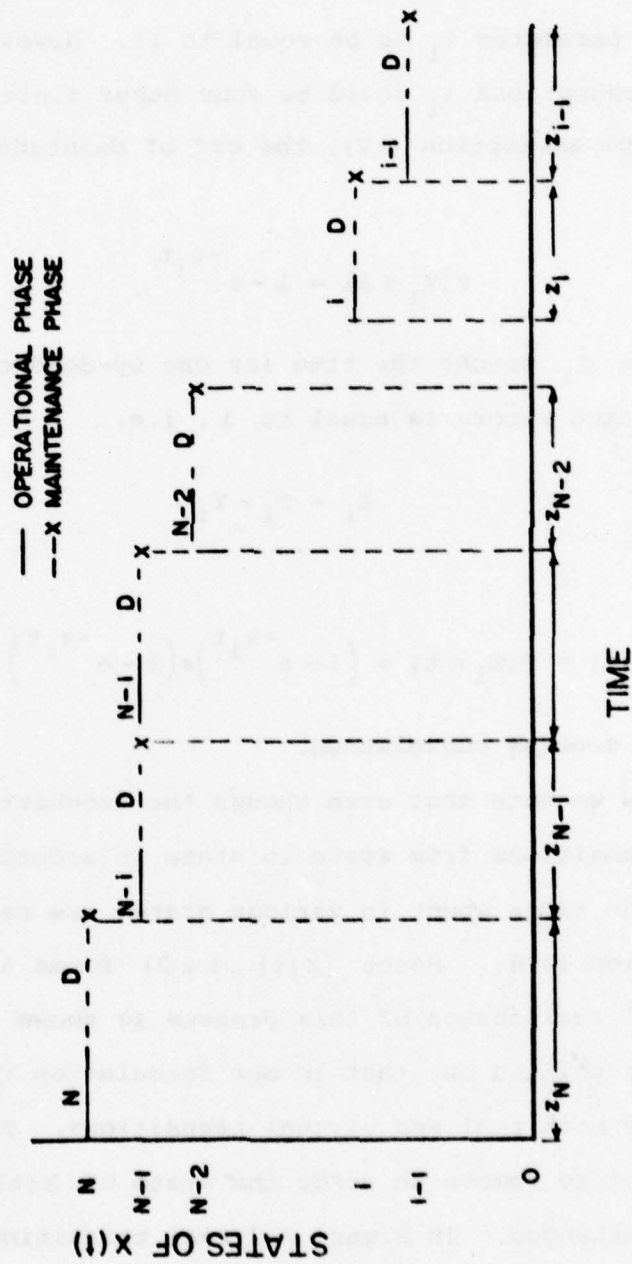


Figure 2.2 A Typical Realization of the $X(t)$ Process

Let $Q_{ij}^{(D)}(t)$ denote the one step transition probability that after making a transition into state i , the process $X(t)$ next makes a transition into state j via D , by time t . In other words if a software package has i remaining errors at time zero, then $Q_{ij}^{(D)}(t)$ represents the probability that the next up-down cycle, resulting in j remaining errors, will be completed by time t . Hence, for $i, j = 0, 1, 2, \dots, N$, we can write

$$Q_{ij}^{(D)}(t) = \int_0^t P\{X(u) = j, Z_i = u | X(0) = i\} \cdot du.$$

Since the events $\{X(u) = j\}$ and $\{Z_i = u\}$ are independent, we get

$$\begin{aligned} Q_{ij}^{(D)}(t) &= \int_0^t P\{X(u) = j | X(0) = i\} \cdot P\{Z_i = u | X(0) = i\} \cdot du \\ &= \int_0^t P_{ij}^{(D)} \cdot P\{Z_i = u | X(0) = i\} \cdot du \\ &= P_{ij}^{(D)} \int_0^t dP(Z_i \leq u) \end{aligned}$$

$$\text{or } Q_{ij}^{(D)}(t) = P_{ij}^{(D)} \xi_i(t) \quad (2.9)$$

for $i, j = 0, 1, 2, \dots, N$.

It is obvious that $Q_{ij}^{(D)}(t)$ must satisfy

$$Q_{ij}^{(D)}(t) \geq 0, \quad i, j = 0, 1, 2, \dots, N, \quad t \geq 0$$

and

$$\sum_{j=0}^N Q_{ij}^{(D)}(\infty) = p+q = 1, \quad i = 0, 1, \dots, N.$$

Now, the one-step transition probabilities from state i to i and to $(i-1)$ via D are obtained from (2.9) and (2.3) as

$$Q_{i,i}^{(D)}(t) = q \cdot \xi_i(t) \quad (2.10)$$

$$Q_{i,i-1}^{(D)}(t) = p \cdot \xi_i(t) \quad (2.11)$$

for $i=1,2,\dots,N$

and

$$Q_{0,0}^{(D)}(t) = 1. \quad (2.12)$$

Proceeding similarly for all i,j , we get

$$\{Q_{ij}^{(D)}(t)\} = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & \dots & N-2 & N-1 & N \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ \vdots \\ \vdots \\ N-1 \\ N \end{matrix} & \left[\begin{array}{ccccccc} 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ p\xi_1(t) & q\xi_1(t) & 0 & \dots & 0 & 0 & 0 \\ 0 & p\xi_2(t) & q\xi_2(t) & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & \dots & p\xi_{N-1}(t) & q\xi_{N-1}(t) & 0 & 0 \\ 0 & 0 & \dots & \dots & p\xi_N(t) & q\xi_N(t) & \end{array} \right] \end{matrix} \quad (2.13)$$

For known parameters N, p, λ_i and μ_i , the probabilities $Q_{ij}^{(D)}(t)$ can be calculated from equation (2.13). This equation is the basic model for the process under study.

3. DERIVATION OF VARIOUS QUANTITIES OF INTEREST

3.1 Distribution of Time to a Specified Number of Remaining Errors

Suppose at some time during the operational phase the number of remaining errors in the software system is i . Let $g_{i,n_0}(t)$ and $G_{i,n_0}(t)$ be the pdf and cdf, respectively, of the first passage time from i to n_0 . These quantities are the pdf and cdf of the time required to obtain a software system with n_0 errors when the initial number of errors is i .

Since the number of errors at time zero is N , we are interested in getting an expression for $G_{N,n_0}(t)$. First consider the case of perfect maintenance. From the definition of $Q_{i,j}^{(D)}(t)$, the probability of going from N to $N-1$ errors via D in time $[u, u+du]$ is $dQ_{N,N-1}^{(D)}(u)$. The process restarts with $(N-1)$ remaining errors at time u and the cdf of first passage time from $N-1$ to n_0 is then $G_{N-1,n_0}(t-u)$. Thus the cdf of first passage time from N to n_0 when the maintenance is perfect is

$$\int_0^t G_{N-1,n_0}(t-u) \cdot dQ_{N,N-1}^{(D)}(u) = Q_{N,N-1}^{(D)} * G_{N-1,n_0}(t). \quad (3.1)$$

Similarly, if the maintenance at the first error removal is imperfect, the cdf of first passage time from N to n_0 is

$$\int_0^t G_{N,n_0}(t-u) \cdot dQ_{N,N}^{(D)}(u) = Q_{N,N}^{(D)} * G_{N,n_0}(t). \quad (3.2)$$

Since the events depicted in (3.1) and (3.2) are mutually exclusive, we get the cdf of first passage time from N to n_0 as

$$G_{N,n_0}(t) = Q_{N,N-1}^{(D)} * G_{N-1,n_0}(t) + Q_{N,N}^{(D)} * G_{N,n_0}(t). \quad (3.3)$$

In general, the renewal equation is

$$G_{i,n_0}(t) = Q_{i,i-1}^{(D)} * G_{i-1,n_0}(t) + Q_{i,i}^{(D)} * G_{i,n_0}(t) \quad (3.4)$$

for $i = n_0+1, n_0+2, \dots, N$ where $G_{n_0,n_0} = 1$.

Using Laplace-Stieltjes (L-S) transforms to solve the renewal equation (3.4), we get

$$\tilde{G}_{i,n_0}(s) = \tilde{Q}_{i,i-1}^{(D)}(s) \tilde{G}_{i-1,n_0}(s) + \tilde{Q}_{i,i}^{(D)}(s) \tilde{G}_{i,n_0}(s) \quad i = n_0+1, n_0+2, \dots, N \quad (3.5)$$

where

$$\begin{aligned} \tilde{G}_{i,n_0}(s) &= \int_0^\infty e^{-st} \cdot dG_{i,n_0}(s) \\ \tilde{Q}_{i,i}^{(D)}(s) &= \frac{q \lambda_i \mu_i}{(s + \lambda_i)(s + \mu_i)}, \end{aligned} \quad (3.6)$$

and

$$\tilde{Q}_{i,i-1}^{(D)}(s) = \frac{p \lambda_i \mu_i}{(s + \lambda_i)(s + \mu_i)}. \quad (3.7)$$

Solving the set of equations (3.5) recursively, we get

$$\begin{aligned} \tilde{G}_{N,n_0}(s) &= \prod_{i=n_0+1}^N \left\{ \frac{p \lambda_i \mu_i}{s^2 + (\lambda_i + \mu_i)s + p \lambda_i \mu_i} \right\} \\ &= \prod_{i=n_0+1}^N \left(\frac{r_{1,i}}{s + r_{1,i}} \right) \left(\frac{r_{2,i}}{s + r_{2,i}} \right) \end{aligned} \quad (3.8)$$

where $r_{1,i}$ and $r_{2,i}$ satisfy

$$r_{1,i} + r_{2,i} = \lambda_i + \mu_i \quad (3.9)$$

$$r_{1,i} r_{2,i} = p \lambda_i \mu_i.$$

Now from Corollary A.2 of Appendix A we get

$$\begin{aligned} \tilde{G}_{N,n_0}(s) &= \tilde{H}_{N,n_0}^1(s) \tilde{H}_{N,n_0}^2(s) \\ &= \tilde{H}_{N,n_0,n_0}(s) \end{aligned} \quad (3.10)$$

where we set $c_{1,i} = r_{1,i}$ and $c_{2,i} = r_{2,i}$. Finally, we obtain the first passage time distribution from N to n_0 as

$$\begin{aligned} G_{N,n_0}(t) &= \sum_{i=n_0+1}^N \sum_{j=n_0+1}^N a_{N,i,n_0+1}^1 a_{N,i,n_0+1}^2 \\ &\times \left[1 - \left\{ r_{2,j} e^{-r_{1,i}t} - r_{1,i} e^{-r_{2,j}t} \right\} / (r_{1,i} - r_{2,j}) \right] \end{aligned} \quad (3.11)$$

where a_{N,i,n_0+1}^1 and a_{N,i,n_0+1}^2 are as given in Appendix A.

From (3.11), the pdf of the first passage time is obtained

as

$$\begin{aligned} g_{N,n_0}(t) &= \frac{d}{dt} G_{N,n_0}(t) = \sum_{i=n_0+1}^N \sum_{j=n_0+1}^N a_{N,i,n_0+1}^1 a_{N,j,n_0+1}^2 \\ &\times \frac{r_{1,i} r_{2,j}}{r_{1,i} - r_{2,j}} \left(e^{-r_{2,j}t} - e^{-r_{1,i}t} \right). \end{aligned} \quad (3.12)$$

3.1.1 Mean and variance of the first passage time

The mean and variance of the first passage time, T_{N,n_0} , from N to n_0 are obtained as follows

$$ET_{N,n_0} = \sum_{i=n_0+1}^N \sum_{j=n_0+1}^N a_{N,i,n_0+1}^1 a_{N,j,n_0+1}^2 \times \left(\frac{r_{1,i}}{r_{2,j}} - \frac{r_{2,j}}{r_{1,i}} \right) / (r_{1,i} - r_{2,j}) \quad (3.13)$$

$$ET_{N,n_0}^2 = \sum_{i=n_0+1}^N \sum_{j=n_0+1}^N a_{N,i,n_0+1}^1 a_{N,j,n_0+1}^2 \times \left(\frac{r_{1,i}}{r_{2,j}^2} - \frac{r_{2,j}}{r_{1,i}^2} \right) / (r_{1,i} - r_{2,j}) \quad (3.14)$$

$$\text{Var}(T_{N,n_0}) = ET_{N,n_0}^2 - (ET_{N,n_0})^2. \quad (3.15)$$

3.1.2 Numerical example

For illustration purposes consider the case when $\lambda_i = i\lambda$, $\mu_i = i\mu$, $N=10$, $p=0.9$, $\lambda=0.02$ and $\mu=0.05$.

The pdf and cdf for the first passage time for this example are plotted in Figures 3.1 and 3.2, respectively. These plots are self explanatory.

The mean and variance of the first passage times from $N=10$ to $n_0=0,1,\dots,9$ are computed from equations (3.13) and (3.15) and are given in Table 3.1.

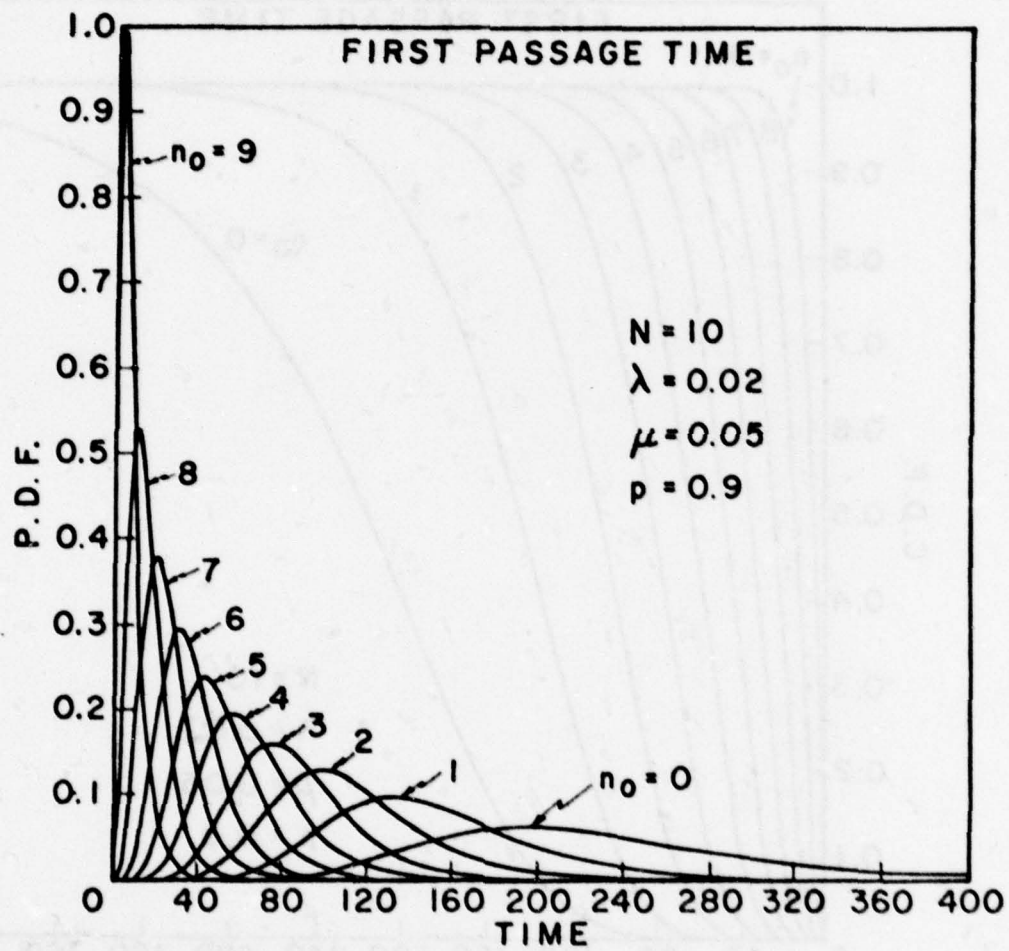


Figure 3.1 PDF of First Passage time from N to n_0

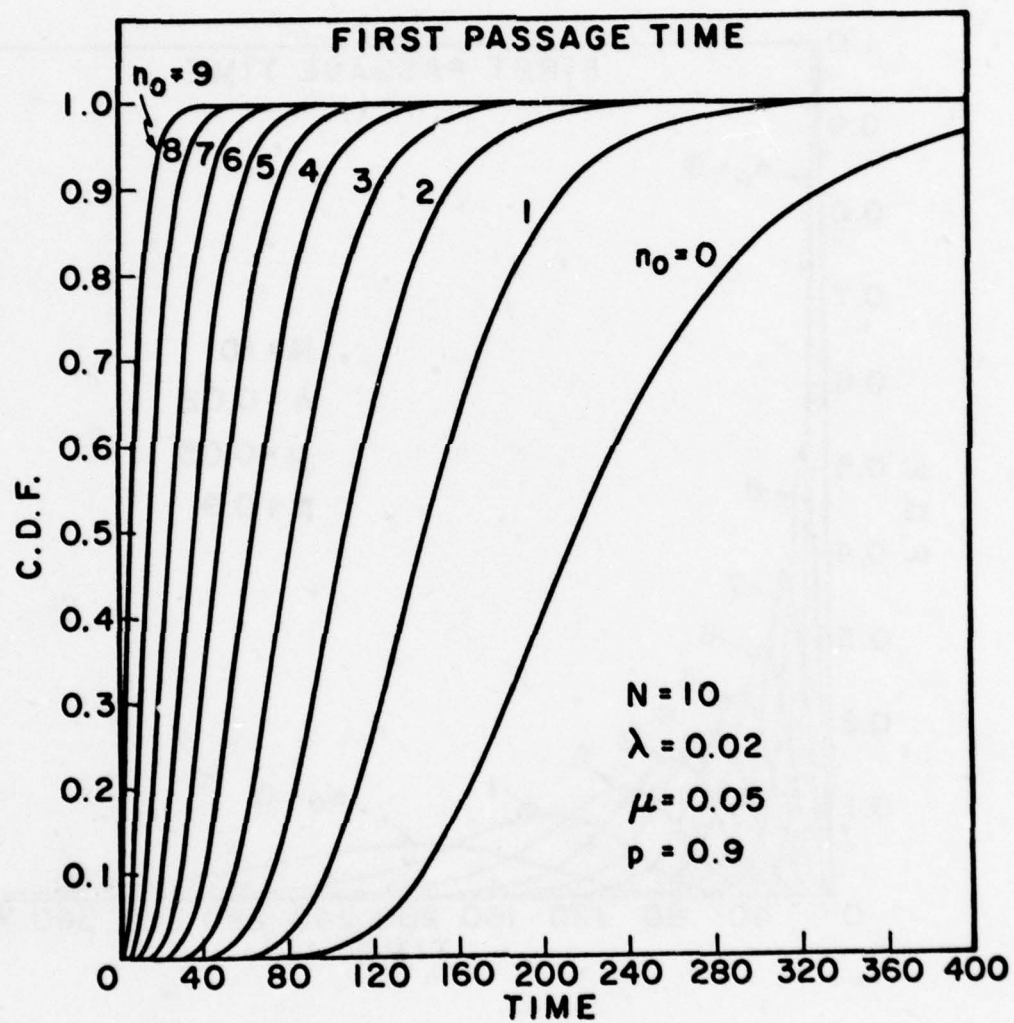


Figure 3.2 CDF of First Passage Time from N to n_0

TABLE 3.1
MEAN AND VARIANCE OF FIRST PASSAGE TIME FOR VARIOUS n_0
($N=10$, $p=0.9$, $\lambda=0.02$, $\mu=0.05$)

n_0	Mean	Variance	$\sqrt{\text{Variance}}$
9	7.78	38.27	6.19
8	16.42	85.52	9.25
7	26.14	145.32	12.05
6	37.25	223.43	14.95
5	50.22	329.74	18.16
4	65.77	482.82	21.97
3	85.22	722.02	26.87
2	111.14	1147.26	33.87
1	150.03	2104.05	45.87
0	227.81	5931.21	77.01

3.2 State Occupancy Probabilities and Software System Availability

In this section we are interested in obtaining expressions for the number of errors remaining in the software system and the system availability. Let

$$P_{N,n_0}(t) = P\{X(t) = n_0 | X(0) = N\}, \quad (3.16)$$

that is, $P_{N,n_0}(t)$ represents the probability that the software system is operational at time t with n_0 remaining errors, given that it was operation at time $t=0$ with N remaining errors.

First we derive the expression for system availability in terms of $P_{N,n_0}(t)$. By conditioning on the first up-down cycle of the process and using an approach similar to that of Section 3.1 we get the following renewal equation:

$$P_{n_0,n_0}(t) = e^{-\lambda_{n_0} t} + Q_{n_0,n_0}^{(D)} * P_{n_0,n_0}(t), \quad n_0 \leq N \quad (3.17)$$

Conditioning on the first passage time, we get

$$P_{N,n_0}(t) = P_{n_0,n_0} * G_{N,n_0}(t). \quad (3.18)$$

Taking the L-S transform of $P_{n_0,n_0}(t)$ in (3.17) and of $P_{N,n_0}(t)$ in (3.18), we get

$$\tilde{P}_{n_0,n_0}(s) = \frac{s}{s + \lambda_{n_0}} + \tilde{Q}_{n_0,n_0}^{(D)}(s) \cdot \tilde{P}_{n_0,n_0}(s) \quad (3.19)$$

and

$$\tilde{P}_{N,n_0}(s) = \tilde{P}_{n_0,n_0}(s) \cdot \tilde{G}_{N,n_0}(s). \quad (3.20)$$

Substituting the L-S of $Q_{n_0, n_0}^{(D)}(t)$ from (3.7) and rearranging, we get

$$\begin{aligned}\tilde{P}_{n_0, n_0}(s) &= \frac{s(s+\mu_{n_0})}{s^2 + (\lambda_{n_0} + \mu_{n_0})s + p\lambda_{n_0}\mu_{n_0}} \\ &= 1 - \frac{\lambda_{n_0} - r_{2, n_0}}{r_{1, n_0} - r_{2, n_0}} \frac{r_{1, n_0}}{s + r_{1, n_0}} - \frac{r_{1, n_0} - \lambda_{n_0}}{r_{1, n_0} - r_{2, n_0}} \frac{r_{2, n_0}}{s + r_{2, n_0}}. \quad (3.21)\end{aligned}$$

Substituting (3.21) into (3.20) we obtain the L-S transform of state occupancy probability, i.e.

$$\begin{aligned}\tilde{P}_{N, n_0}(s) &= \tilde{G}_{N, n_0}(s) - \frac{\lambda_{n_0} - r_{2, n_0}}{r_{1, n_0} - r_{2, n_0}} \tilde{H}_{N, n_0-1}^1(s) \tilde{H}_{N, n_0}^2(s) \\ &\quad - \frac{r_{1, n_0} - \lambda_{n_0}}{r_{1, n_0} - r_{2, n_0}} \tilde{H}_{N, n_0}^1(s) \tilde{H}_{N, n_0-1}^2(s) \quad (3.22)\end{aligned}$$

where

$$\tilde{H}_{N, -1}^1(s) = \tilde{H}_{N, -1}^2(s) = 0 \quad (3.23)$$

and

$$\tilde{G}_{N, N}(s) = \tilde{H}_{N, N}^1(s) = \tilde{H}_{N, N}^2(s) = 1. \quad (3.24)$$

Note that we use equation (3.10) to obtain (3.22). Therefore, from Corollary A.2 we have

$$\begin{aligned}P_{N, n_0}(t) &= G_{N, n_0}(t) - \frac{\lambda_{n_0} - r_{2, n_0}}{r_{1, n_0} - r_{2, n_0}} H_{N, n_0-1, n_0}(t) \\ &\quad - \frac{r_{1, n_0} - \lambda_{n_0}}{r_{1, n_0} - r_{2, n_0}} H_{N, n_0, n_0-1}(t) \quad n_0 = 0, 1, 2, \dots, N \quad (3.25)\end{aligned}$$

which can be computed from the first passage time distribution

$G_{N, n_0}(t)$.

Let $A(t)$ be the software system availability at time t , that is, the probability that the software system is operational at time t . Then, from the definition of $P_{N,n_0}(t)$ we get the expression for $A(t)$, i.e.

$$A(t) = \sum_{n_0=0}^N P_{N,n_0}(t). \quad (3.26)$$

Figure 3.3 shows the software system availability, $A(t)$ and the state occupancy probabilities, $P_{N,n_0}(t)$, for the case when $N=10$, $p=0.9$, $\lambda=0.02$ and $\mu=0.05$. It shows how the availability improves with time.

Now, let $\bar{N}(t)$ be the number of errors remaining in the software at time t . Then, the distribution of $\bar{N}(t)$ is given by

$$P\{\bar{N}(t)=n\} = G_{N,n}(t) - G_{N,n-1}(t) \quad n=0,1,2,\dots,N \quad (3.27)$$

with mean

$$E\bar{N}(t) = \sum_{n=1}^N [1 - G_{N,n-1}(t)]. \quad (3.28)$$

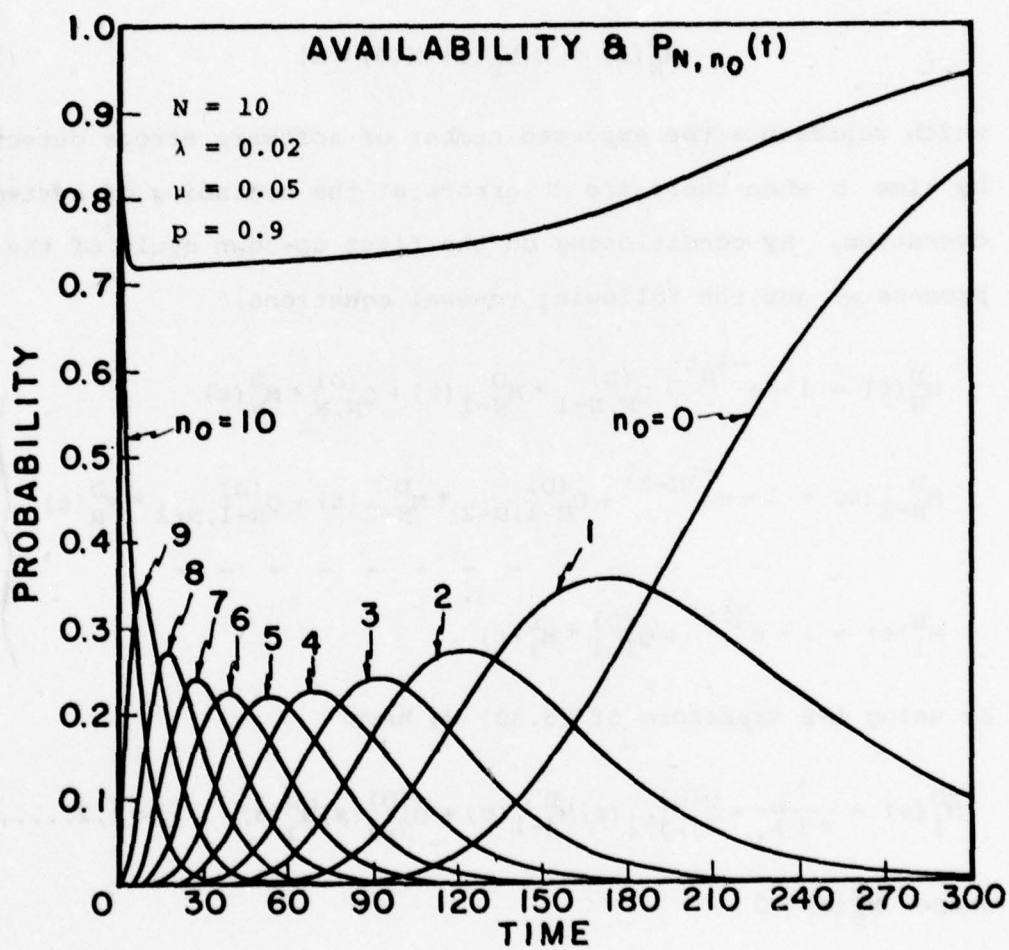


Figure 3.3 Software System Availability and State Occupancy Probabilities at Time t .

3.3 Number of Software Errors Detected by Time t

Let us introduce a counting process $\{N_D(t), t \geq 0\}$, where $N_D(t)$ denotes the number of software errors detected by time t . We are interested in deriving the expression for a renewal function of this counting process. Let

$$M_N^D(t) = E[N_D(t) | X(0) = N] \quad (3.29)$$

which represents the expected number of software errors detected by time t when there are N errors at the beginning of system operation. By conditioning on the first up-down cycle of the process we get the following renewal equations:

$$\left. \begin{aligned} M_N^D(t) &= 1 - e^{-\lambda_N t} + Q_{N,N-1}^{(D)} * M_{N-1}^D(t) + Q_{N,N}^{(D)} * M_N^D(t) \\ M_{N-1}^D(t) &= 1 - e^{-\lambda_{N-1} t} + Q_{N-1,N-2}^{(D)} * M_{N-2}^D(t) + Q_{N-1,N-1}^{(D)} * M_{N-1}^D(t) \\ &\quad - - - - - \\ M_1^D(t) &= 1 - e^{-\lambda_1 t} + Q_{1,1}^{(D)} * M_1^D(t) . \end{aligned} \right\} \quad (3.30)$$

By using L-S transform of (3.30) we have

$$\tilde{M}_j^D(s) = \frac{\lambda_j}{s + \lambda_j} + \tilde{O}_{j,j-1}^{(D)}(s) \tilde{M}_{j-1}^D(s) + \tilde{O}_{j,j}^{(D)}(s) \tilde{M}_j^D(s) \quad j = 1, 2, \dots, N \quad (3.31)$$

where $\tilde{M}_0^D(s) = 0$.

Solving (3.31) recursively and applying the result from equation (3.8), we get

$$\begin{aligned}
\tilde{M}_N^D(s) &= \sum_{i=1}^N \left(\sum_{j=i+1}^N \frac{p\lambda_j \mu_j}{s^2 + (\lambda_j + \mu_j)s + p\lambda_j \mu_j} \right) \frac{\lambda_i s + \lambda_i \mu_i}{s^2 + (\lambda_i + \mu_i)s + p\lambda_i \mu_i} \\
&= \sum_{i=1}^N \tilde{G}_{N,i}(s) \left[\frac{\lambda_i}{r_{1,i} - r_{2,i}} \left\{ \left(1 - \frac{\mu_i}{r_{1,i}}\right) \frac{r_{1,i}}{s + r_{1,i}} + \left(\frac{\mu_i}{r_{2,i}} - 1\right) \frac{r_{2,i}}{s + r_{2,i}} \right\} \right] \\
&= \sum_{i=1}^N \frac{\lambda_i}{r_{1,i} - r_{2,i}} \left\{ \left(1 - \frac{\mu_i}{r_{1,i}}\right) \tilde{H}_{N,i-1}^1(s) \tilde{H}_{N,i}^2(s) \right. \\
&\quad \left. + \left(\frac{\mu_i}{r_{2,i}} - 1\right) \tilde{H}_{N,i}^1(s) \tilde{H}_{N,i-1}^2(s) \right\}. \tag{3.32}
\end{aligned}$$

Then we have a renewal function of this counting process i.e.

$$\begin{aligned}
M_N^D(t) &= \sum_{i=1}^N \frac{\lambda_i}{r_{1,i} - r_{2,i}} \left\{ \left(1 - \frac{\mu_i}{r_{1,i}}\right) H_{N,i-1,i}(t) \right. \\
&\quad \left. + \left(\frac{\mu_i}{r_{2,i}} - 1\right) H_{N,i,i-1}(t) \right\} \tag{3.33}
\end{aligned}$$

where the expressions for $H_{N,i-1,i}(t)$ and $H_{N,i,i-1}(t)$ are given in Corollary A.2.

3.4 Number of Software Errors Corrected by Time t

We now introduce another counting process $\{N_C(t), t \geq 0\}$, where $N_C(t)$ denotes the number of software errors corrected by time t . Also, let

$$M_N^C(t) = E[N_C(t) | X(0) = N] \tag{3.34}$$

which represents the expected number of software errors corrected by time t when there are N errors at the beginning of system operation. If we condition on the first up-down cycle of the process, we can get the following renewal equations:

$$M_j^C(t) = (1 - e^{-\lambda_j t}) * (1 - e^{-\mu_j t}) + Q_{j,j-1}^{(D)} * M_{j-1}^C(t) + O_{j,j}^{(D)} * M_j^C(t) \quad (3.35)$$

$$j = 1, 2, \dots, N,$$

where $M_0^C(t) = 0$.

Solving (3.35) recursively and applying the result from equation (3.8) we get

$$\begin{aligned} \tilde{M}_N^C(s) &= \frac{1}{p} \sum_{i=1}^N \prod_{j=i}^N \frac{p \lambda_j \mu_j}{s^2 + (\lambda_j + \mu_j)s + p \lambda_j \mu_j} \\ &= \frac{1}{p} \sum_{i=1}^N \tilde{G}_{N,i-1}(s). \end{aligned} \quad (3.36)$$

Finally, we get the expression for $M_N^C(t)$ in terms of first passage time distribution i.e.

$$M_N^C(t) = \frac{1}{p} \sum_{i=1}^N G_{N,i-1}(t). \quad (3.37)$$

Comparing equations (3.33) and (3.37), the expression for $M_N^C(t)$ differs only slightly from the one for $M_N^D(t)$ because of the time to maintain a software failure involved in this model. However, as $t \rightarrow \infty$ we have

$$M_N^D(\infty) = M_N^C(\infty) = \frac{N}{p}. \quad (3.38)$$

For illustration purposes consider the case when $N = 10$, $p = 0.9$, $\lambda = 0.02$ and $\mu = 0.05$. The results for $M_N^D(t)$ and $M_N^C(t)$ are plotted in Figure 3.4. It is easy to see that the difference between the two curves gets small as t gets large. However, if μ gets large, i.e., the maintenance time goes up, the two curves will get further apart.

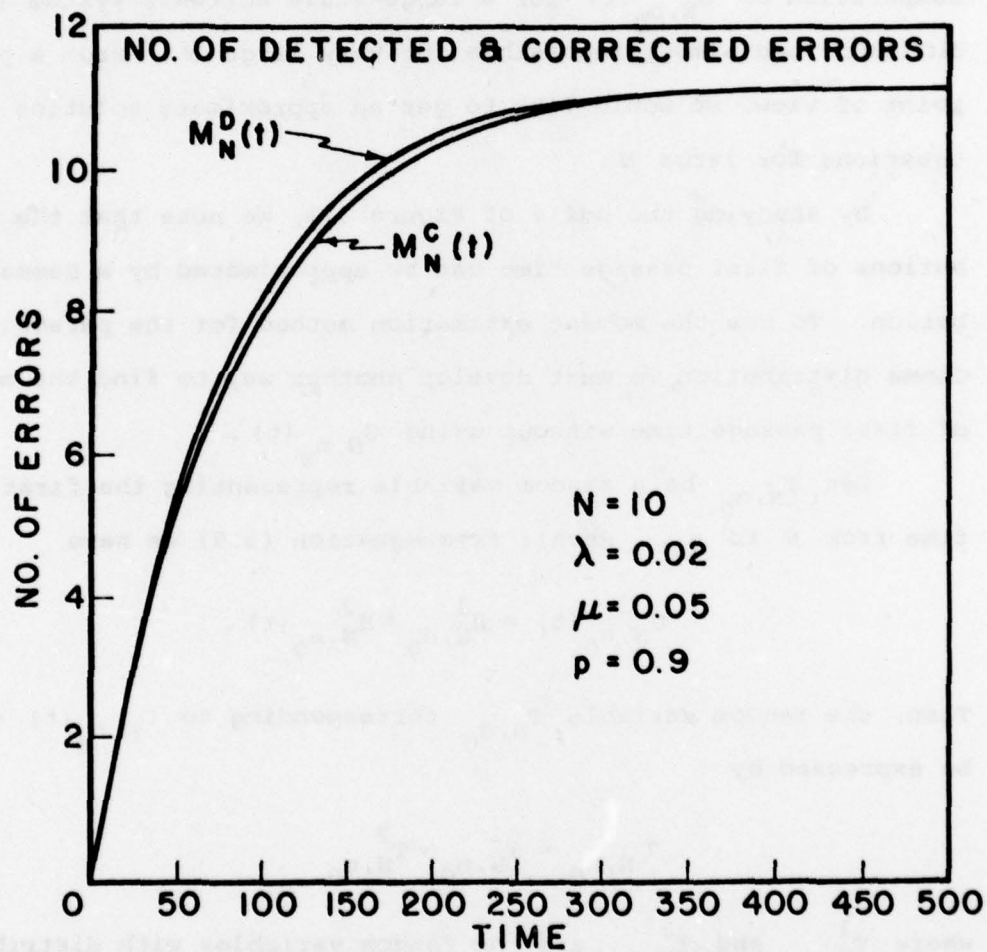


Figure 3.4 Expected Number of Software Errors Detected and Corrected by Time t

4. GAMMA APPROXIMATION FOR A LARGE-SCALE SOFTWARE SYSTEM

In section 3 we obtained several quantities of interest in terms of first passage time distribution, $G_{N,n_0}(t)$. However, the computation of $G_{N,n_0}(t)$ for a large-scale software system is difficult and almost impossible for very large N . From a practical point of view, we would like to get an approximate solution of these equations for large N .

By studying the pdf's of Figure 3.1, we note that the distributions of first passage time can be approximated by a Gamma distribution. To use the moment estimation method for the parameters of a Gamma distribution we must develop another way to find the moments of first passage time without using $G_{N,n_0}(t)$.

Let T_{N,n_0} be a random variable representing the first passage time from N to n_0 . Recall from equation (3.9) we have

$$G_{N,n_0}(t) = H_{N,n_0}^1 * H_{N,n_0}^2(t). \quad (4.1)$$

Then, the random variable T_{N,n_0} corresponding to $G_{N,n_0}(t)$ can be expressed by

$$T_{N,n_0} = T_{N,n_0}^1 + T_{N,n_0}^2 \quad (4.2)$$

where T_{N,n_0}^1 and T_{N,n_0}^2 are the random variables with distributions $H_{N,n_0}^1(t)$ and $H_{N,n_0}^2(t)$, respectively. Also note that from the definitions of $H_{N,n_0}^1(t)$ and $H_{N,n_0}^2(t)$ they are in the same form of first passage distribution discussed in Goel and Okumoto [7].

By using the results obtained in [7], we get

$$ET_{N,n_0}^1 = \sum_{i=n_0+1}^N 1/r_{1,i} \quad (4.3)$$

$$ET_{N,n_0}^2 = \sum_{i=n_0+1}^N 1/r_{2,i} \quad (4.4)$$

$$\text{Var}(T_{N,n_0}^1) = \sum_{i=n_0+1}^N 1/r_{1,i}^2 \quad (4.5)$$

$$\text{Var}(T_{N,n_0}^2) = \sum_{i=n_0+1}^N 1/r_{2,i}^2. \quad (4.6)$$

Then, from (4.2) we get the mean and variance of T_{N,n_0} , i.e.

$$ET_{N,n_0} = \sum_{i=n_0+1}^N (1/r_{1,i} + 1/r_{2,i}) \quad (4.7)$$

$$\text{Var}(T_{N,n_0}) = \sum_{i=n_0+1}^N (1/r_{1,i}^2 + 1/r_{2,i}^2). \quad (4.8)$$

Suppose that the Gamma distribution to be used as an approximation for $G_{N,n_0}(t)$ has shape parameter α and scale parameter β , so the mean and variance are given by α/β and α/β^2 , respectively. Using the method of moments for estimating the parameters α and β we have

$$\sum_{i=n_0+1}^N (1/r_{1,i} + 1/r_{2,i}) = \alpha/\beta \quad (4.9)$$

$$\sum_{i=n_0+1}^N (1/r_{1,i}^2 + 1/r_{2,i}^2) = \alpha/\beta^2. \quad (4.10)$$

Therefore, we have

$$\hat{\beta} = \frac{\sum_{i=n_0+1}^N (1/r_{1,i} + 1/r_{2,i})}{\sum_{i=n_0+1}^N (1/r_{1,i}^2 + 1/r_{2,i}^2)} \quad (4.11)$$

$$\hat{\alpha} = \frac{\left\{ \sum_{i=n_0+1}^N (1/r_{1,i} + 1/r_{2,i}) \right\}^2}{\sum_{i=n_0+1}^N (1/r_{1,i}^2 + 1/r_{2,i}^2)} \quad (4.12)$$

Numerical examples of this approximation for various n_0 are given in Table 4.1, where $N=100$, $p=0.9$, $\lambda=0.02$ and $\mu=0.05$. We also compute the relative losses for 3rd, and 4th moments around the mean, to see how good the approximations are. Figure 4.1 shows the relative losses for 3rd and 4th moments around the mean with N , where $p=0.9$, $\lambda=0.02$, $\mu=0.05$ and $n_0=0.2N$.

Based on these results we find that the Gamma approximation of first passage time distribution for a large-scale software system is reasonably good.

Plots of first passage time using Gamma approximations are shown in Figure 4.2 for $N=100$, $p=0.9$, $\lambda=0.02$ and $\mu=0.05$.

The state occupancy probabilities and software system availability based on Gamma approximation for $N=100$, $p=0.9$, $\lambda=0.02$ and $\mu=0.05$ are given in Figure 4.3.

TABLE 4.1

GAMMA APPROXIMATIONS FOR FIRST PASSAGE TIME DISTRIBUTIONS

(N=100, p=0.9, $\lambda=0.02$, $\mu=0.05$)

n_0	Mean	Var	$\hat{\alpha}$	$\hat{\beta}$	Relative Loss (%)	
					3rd. Moment	4th Moment
10	175.65	326.14	94.61	0.539	35.95	4.17
15	145.38	208.75	101.25	0.696	30.09	2.86
20	123.64	148.57	102.89	0.832	25.84	2.19
25	106.67	111.98	101.60	0.953	22.61	1.80
30	92.74	87.39	98.42	1.061	20.10	1.55
35	80.94	69.72	93.96	1.161	18.11	1.39
40	70.69	56.41	88.57	1.253	16.52	1.28
45	61.63	46.03	82.53	1.339	15.24	1.22
50	53.52	37.70	75.99	1.420	14.21	1.19
55	46.18	30.88	69.08	1.496	13.38	1.20
60	39.47	25.18	61.89	1.568	12.72	1.23
70	27.58	16.20	46.93	1.702	11.77	1.43
80	17.26	9.46	31.48	1.824	11.22	1.93
90	8.15	4.21	15.79	1.94	10.93	3.45

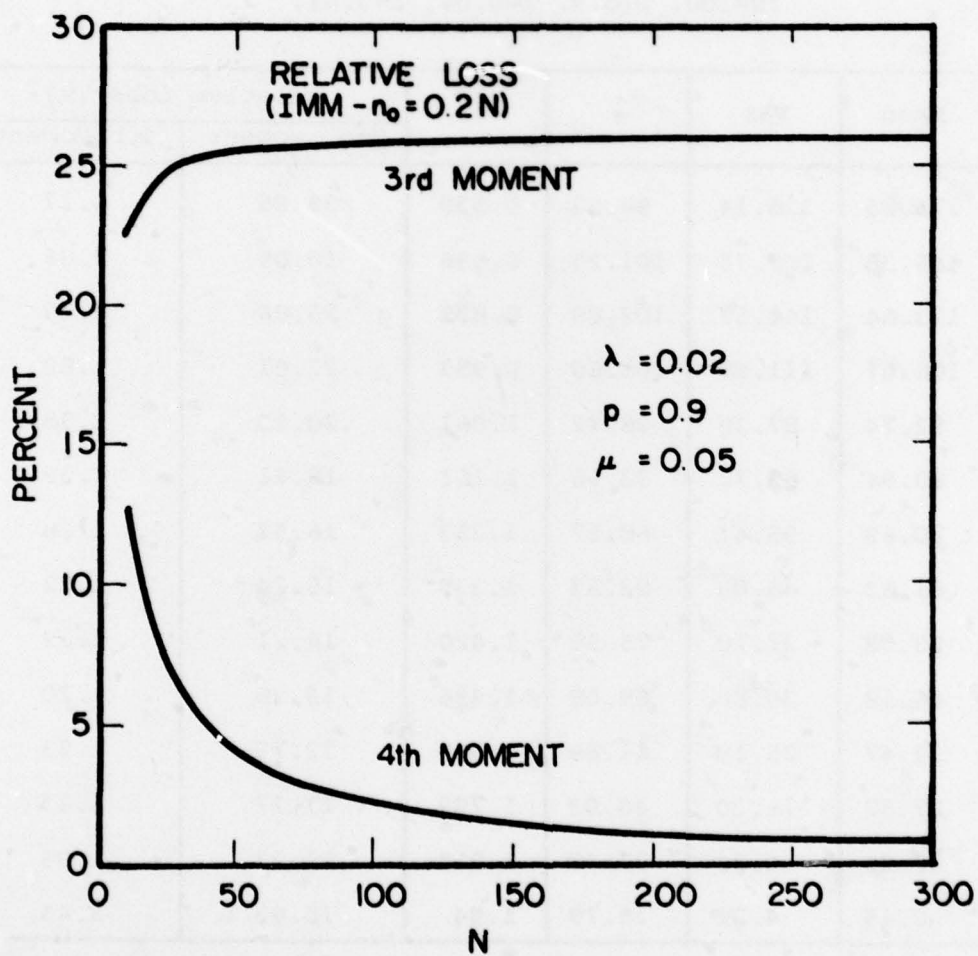


Figure 4.1 Relative Losses for the Third and Fourth Moments

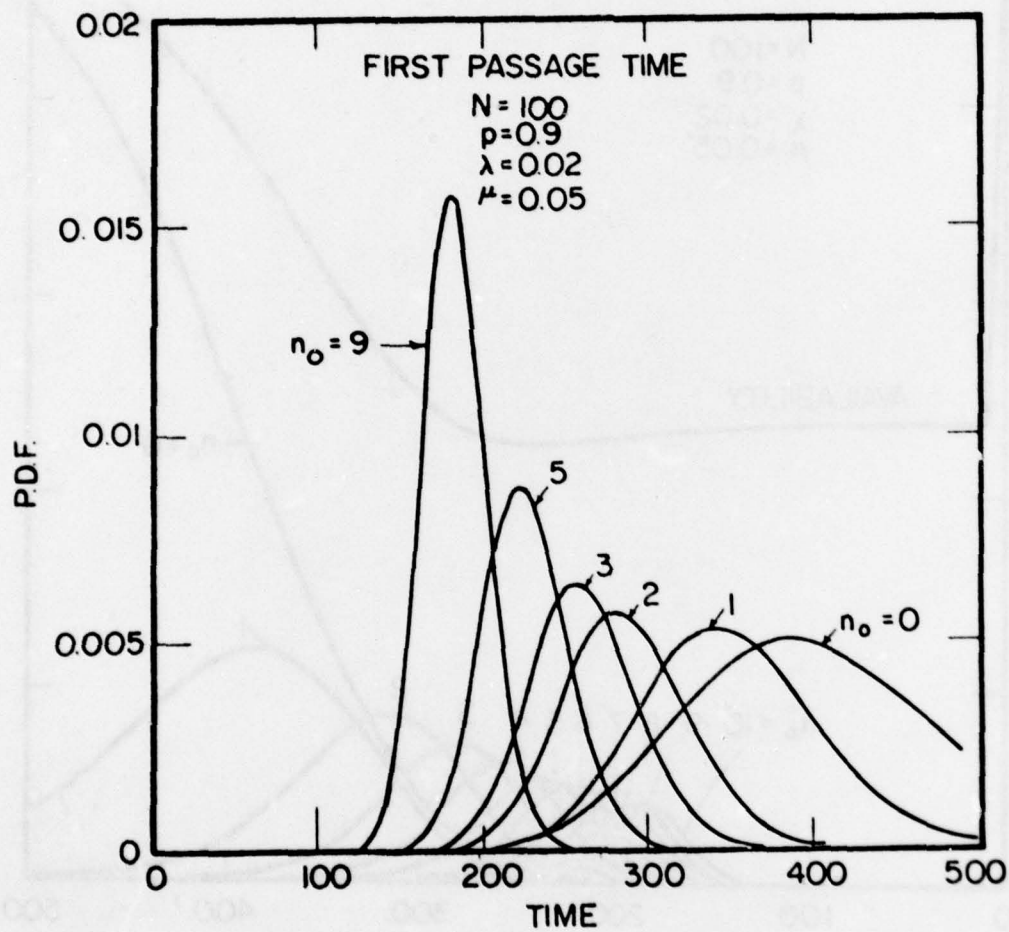


Figure 4.2 First Passage Times Based
on Gamma Approximation

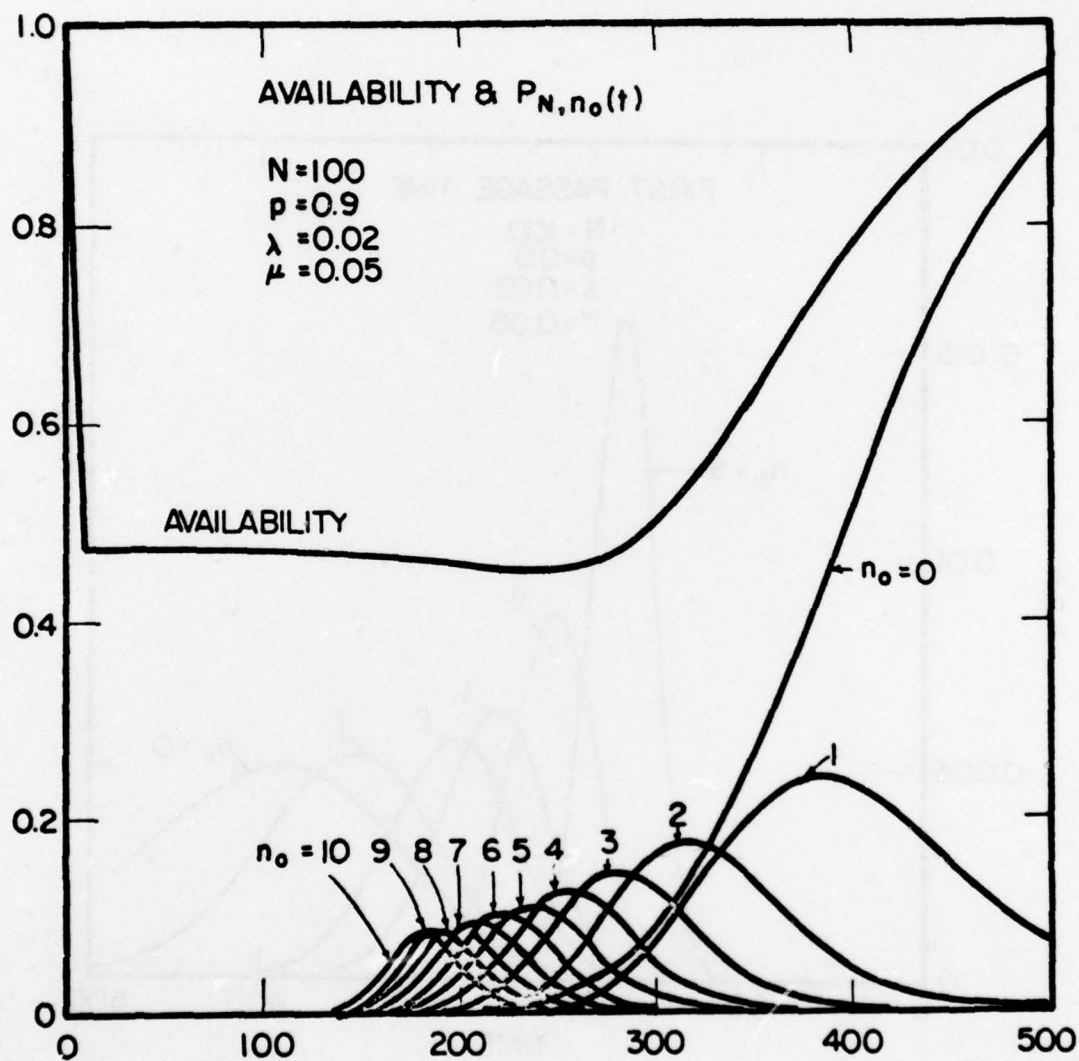


Figure 4.3 Plots of State Occupancy Probabilities and Software System Availability Based on Gamma Approximation

5. CONCLUDING REMARKS

In Section 2 we developed a model for the operational phase of a software system which incorporates the uncertainty of error removal and the time spent in correcting errors.

In Section 3, expressions were derived for the distribution of time to a specified number (n_0) of errors remaining in the software system starting with an initial number of errors N , the state occupancy probabilities, software system availability, and the number of errors detected and corrected by time t .

An approximation using Gamma distribution was discussed in Section 4 for large-scale software systems.

6. SELECTED REFERENCES

- [1] Akiyama, F. (1971), "An Example of Software Debugging," 1971 IFIP Congress, pp. TA-3-37 to TA-3-42.
- [2] Baker, W. F. (1977), "Software Data Collection and Analysis: A Real-Time System Project History," IBM Corporation, Final Technical Report, RADC-TR-77-192, June 1977.(A041644)
- [3] Belady, L. A. and Lehman, M. M. (1976), A Model of Large Program Development, IBM Systems Journal, Vol. 15, no. 3 pp. 225-252.
- [4] Boehm, B. W., Brown, J. R., and Lipow, M. (1976), "Quantitative Evaluation of Software Quality," Proc. 2nd International Conference on Software Engineering, pp. 54-59.
- [5] Endres, A. (1975), "An Analysis of Errors and Their Causes in System Programs," Proceedings: 1975 International Conference on Reliable Software, pp. 327-336.
- [6] Fries, M. J. (1977), "Software Error Data Acquisition," Boeing Aerospace Company, Final Technical Report, RADC-TR-77-130, April 1977.(A039916)
- [7] Goel, A. L. and Okumoto, K. (1978), "An Imperfect Debugging Model for Reliability and Other Quantitative Measures of Software Systems," Technical Report No. 78-1, Department of IE & OR, Syracuse University.
- [8] Goel, A. L. and Okumoto, K. (1978), "Software Failure Analysis by Non-Homogeneous Poisson Process,, Technical Report No. 78-7, Department of IE & OR, Syracuse University.
- [9] Halstead, M. H. (1977), Elements of Software Science, Elsevier.
- [10] Jelinski, J. and Moranda, P. B. (1972), "Software Reliability Research," in Statistical Computer Performance Evaluation, edited by W. Freiberger, Academic press.
- [11] Jelinski, J. and Moranda, P. B. (1973), "Applications of a Probability-Based Model to a Code Reading Experiment," Record: 1973 IEEE Symposium on Computer Software Reliability, pp. 78-81.
- [12] Littlewood, B. (1975), "A Reliability Model for Markov Structured Software," Applied Statist., Vol. 24, pp. 172-177.

- [13] Littlewood, B. and Verrall, J. L. (1973), "A Bayesian Reliability Growth Model for Computer Software," Applied Statist., Vol. 22, pp. 332-346.
- [14] Miyamoto, I. (1975), "Software Reliability in On-Line Real Time Environment," Proceedings: 1975 International Conference on Reliable Software, pp. 194-203.
- [15] Moranda, P. (1975), "A Comparison of Software Error-rate Models," 1975 Texas Conference on Computing.
- [16] Motley, R. W. et al (1977), "Statistical Prediction of Programming Errors," IBM Corporation, Final Technical Report, RADC-TR-77-175, May 1977. (A041106)
- [17] Musa, J. D. (1975), "A Theory of Software Reliability and its Application," IEEE Trans. on Software Engineering, Vol. SE-1, No. 3, pp. 312-327.
- [18] Ross, S. M. (1970), Applied Probability Models With Optimization Applications, Holden-Day.
- [19] Roussas, G. G. (1973), A First Course in Mathematical Statistics, Addison-Wesley.
- [20] Rye, P. et al (1977), "Software Systems Development: A CSDL Project History," The Charles Stark Draper Laboratory, Inc., Final Technical Report, RADC-TR-77-213, June 1977. (A042186)
- [21] Schick, G. J. and Wolverton, R. W. (1972), "Assessment of Software Reliability," McDonnell-Douglas Astronautics Company Paper WD1872.
- [22] Schneidewind, N. J. (1972), "An Approach to Software Reliability Prediction and Quality Control," AFIPS Conference Proceedings, Vol. 41 Part II, Fall Joint Computer Conference, pp. 837-838.
- [23] Schneidewind, N. J. (1975), "Analysis of Error Processes in Computer Software," Proceedings: 1975 International Conference on Reliable Software, pp. 337-346.
- [24] Shooman, M. L. (1972), "Probabilistic Models for Software Reliability Prediction," Statistical Computer Performance Evaluation, pp. 485-502, Academic Press, New York.
- [25] Shooman, M. L. and Bolsky, M. I. (1975), "Types, Distribution, and Test Correction Times for Programming Errors,"

Proceedings: 1975 International Conference on Reliable Software, pp. 347-357.

- [26] Sukert, A. N. (1977), "An Investigation of Software Reliability Models," Proc. 1977 R & M.
- [27] Sukert, A. N. (1977), "A Multi-Project Comparison of Software Reliability Models," Computers in Aerospace Conference, pp. 413-421.
- [28] Thayer, T. A. et al (1976), "Software Reliability Study," TRW Defense & Space Systems Group, Final Technical Report, RADC-TR-76-238, Aug. 1976. (A030798)
- [29] Trivedi, A. K. and Shooman, M. L. (1975), "A Many-State Markov Model for the Estimation and Prediction of Computer Software Performance Parameters," Proceedings: 1975 International Conference on Reliable Software, pp. 208-220.
- [30] Wagoner, W. L. (1973), "The Final Report on Software Reliability Measurement Study," Aerospace Report No. TOR-0074(4112)-1.
- [31] Willman, H. E. Jr. et al (1977), "Software Systems Reliability: A Raytheon Project History," Raytheon Company, Final Technical Report, RADC-TR-77-188, June 1977. (A040992)

APPENDIX A

The following lemma which is known as Heaviside Expansion Theorem is useful in our analysis.

Lemma A.1 For any integer values n_0 , $N \geq n_0$ and any real number $c_i > 0$, $i = n_0, n_0+1, \dots, N$

$$\prod_{i=n_0}^N \frac{c_i}{s + c_i} = \sum_{i=n_0}^N a_{N,i,n_0} \frac{c_i}{s + c_i} \quad (\text{A.1})$$

where

$$a_{N,i,n_0} = \prod_{\substack{j=n_0 \\ j \neq i}}^N \frac{c_j}{c_j - c_i}. \quad (\text{A.2})$$

From Lemma A.1 we give the following corollary which is needed for computational purposes:

Corollary A.2 Let

$$\tilde{H}_{N,n_1,n_2}(s) = \tilde{H}_{N,n_1}^1(s) \tilde{H}_{N,n_2}^2(s) \quad (\text{A.3})$$

where

$$\tilde{H}_{N,n_1}^1(s) = \prod_{i=n_1+1}^N \frac{c_{1,i}}{s + c_{1,i}} \quad (\text{A.4})$$

$$\tilde{H}_{N,n_2}^2(s) = \prod_{i=1}^N \frac{c_{2,i}}{s + c_{2,i}}. \quad (\text{A.5})$$

Then the inverse of Laplace-Stieltjes (L-S) transform of $\tilde{H}_{N,n_1,n_2}(s)$ is given by

$$H_{N,n_1,n_2}(t) = \sum_{i=n_1+1}^N \sum_{j=n_2+1}^N a_{N,i,n_1+1}^1 \cdot a_{N,j,n_2+1}^2 \\ \times \left[1 - \left\{ c_{2,j} e^{-c_{1,i}t} - c_{1,i} e^{-c_{2,j}t} \right\} / (c_{1,i} - c_{2,j}) \right] \quad (A.6)$$

where a_{N,i,n_1+1}^1 and a_{N,j,n_2+1}^2 are the coefficients corresponding to $\tilde{H}_{N,n_1}^1(s)$ and $\tilde{H}_{N,n_2}^2(s)$, respectively.

The proof of Corollary A.2 is obvious.